California State Polytechnic University, Pomona

ME 5741 – Biomechanical Robots

Project #2: Design and Fabrication of Actuating and Grasping System

Project Report

CAL POLY POMONA

By: Ryan Chao, Cristian Rodriguez, and Cameron Weigel

To: Dr. Carlos Castro

Due Date: April 11th, 2022

Date of Submittal: April 11th, 2022

Abstract

The goal of this project was to design and fabricate a biomechanical gripper of up to four fingers. This gripper would be tested on various objects weighing up to 1 pound, having a length up to four inches, and of arbitrary stiffness. Additionally, the gripper shall autonomously grip a target object within 8 seconds of initiation without damaging or dropping the item for at least 45 seconds. We successfully created a two finger, underactuated, compliant gripper with force sensitive feedback to meet the requirements of the project. The gripper was successful in gripping the three objects that were tested in class: a 1 pound, 2.8" diameter stainless steel water bottle, a 0.07 oz 1.8" diameter, 3.4" length tube balloon, and a 2.6 oz, 2.4" diameter empty water bottle with an aluminum foil leaf and tube assembly that was sensitive to overpressure. Our gripper succeeded in holding all three test objects, although some improvements could be made. The most critical improvement to our gripper would be the addition of sensors throughout each segment of the fingers. With sensors integrated only in the fingertips, certain grasping methods would not activate the sensors soon enough or at all. Despite this issue, the gripper was still successful in completing the test.

Introduction

The hand is an extremely complex part of the human anatomy, capable of expressive dexterous articulation. It allows for a wide range of applications such as typing on a keyboard, playing the violin, or communicating via sign language. Each represents one of the six types of grasping methods – cylindrical grasp, tip, hook, palmar, spherical, and lateral [1]. The ability for the hand to develop, recognize, and perform these types of movements makes the process of manipulation with such precision seem effortless. Furthermore, the compliant nature of the skin permits conforming contact points, which results in better prehension [2]. If one were to try and replicate the actuation of the human hand, first, one would need to construct both a rigid, but also compliant mechanical system. Second, the mechanical design would need a minimum of 27 degrees of freedom, which are representative of the total amount of articulations, or joints, present in the hand. Lastly, a feedback system shall be put in place that is representative of the mechanoreceptors found naturally in the skin of the hand. These basic requirements allow for the development of a grasping system that can achieve the complexity of the human hand without adhering verbatim to the design principles of its biological counterpart.

The development of a bioinspired grasping system based on the human hand presents a great advantage in multiple fields, such as industrial soft robotics, humanoid robotic development, and prosthetics. In the field of prosthetics, an actuating gripping system with sensory feedback will eventually return the feeling and capability of a biological human hand back to those suffering from missing limbs at birth or amputees. Sensory feedback in this instance is defined as the capability of a sensor to measure applied force, temperature, vibration, and other physical properties by interaction with its environment [3]. For the development of an actuating prosthetic hand, effective and precise force output regulation is a major design challenge that shall be addressed and solved. For this reason, tactile sensing, has garnered the most attention amongst the sensory feedback mediums in the design of prosthetic hands. Another design challenge that shall be tackled for prosthetic development is the need for the prosthetic to meet mass and size limitations based on those set forth by the average size and weight of biological limbs. It should be noted that the way the human hand meets these requirements is by use of a tendon driven system with fibrous muscles driving the tendons. Meeting these requirements from a prosthetic standpoint, several different actuating systems have been developed, such as linkage systems, gear systems, pneumatic systems, and even micro electro-mechanical systems, or MEMS, to grant a compact design. It remains difficult, however, to incorporate all these factors into the development of a biomimetic robotic hand for upper limb prosthesis. The development of such a convoluted system requires a step-by-step approach beginning with the most common manipulation of the hand – grasping. This report discusses the design and fabrication of a 3D printed soft-rigid tendon driven gripper with force sensing technology.

System Design

The requirements for the grasping system developed in this report are as follows. First, the system shall hold a target object weighing up to 1 lbs. or around 454 grams. Second, the system shall autonomously grasp the target object within 8 seconds from its initial movement. Third, the system shall hold the target object for a minimum of 45 seconds. Fourth, the system shall hold a target object of arbitrary stiffness without damaging the object. Fifth, the system shall be able to hold an object with a maximum dimension of up to 4 inches. Last, the system shall have "fingers", but shall have no more than four. To meet the requirements of the system, three different functional groups of the system were defined: actuation, tactile sensing, and software.

ACTUATION

The method of actuation chosen for this system design was tendon driven. The advantages of a tendon driven system are that it is lightweight, it is compliant, and it is simple in implementation. We found inspiration for our design in a recently published paper titled "Compliant gripper design, prototyping and modeling using screw theory" published in the International Journal of Robotics Research in August of 2020 [4]. The author, Irfan Hussain from the Khalifa University Center for Autonomous Robotic Systems in Abu Dhabi, UAE, and his team developed an under actuating compliant gripper to be used in soft robotics applications.

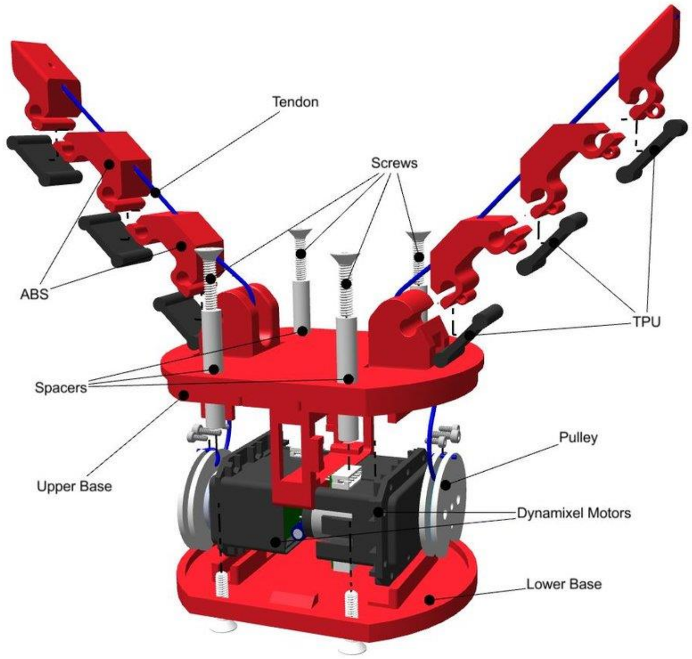

Figure 1. Compliant tendon driven system from Khalifa University

The CAD model for which the system design of this report is based on is found in Figure 1. The Hussain gripper design from a mechanical perspective consists of a very simple cable-pulley system to drive the tendon actuation. An interesting feature of this design that is worth going in depth about are the TPU joints that connect the finger joints. These black TPU pieces are flexible 3D printed parts that not only hold the finger joints together, but also provide a radius of curvature when tension is applied to the cable via pulley and return the fingers back to their initial position when tension is released. Because the TPU pieces were 3D printed, the percentage of the print infill and orientation of the print produced a variable compliance of the gripping system. This variable compliance can, therefore, be used to create different gripping configurations for the system between power and pinch grasps. The stiffer the joint is in this design, the more the system moves towards a pinch grasp and vice versa. The dynamic equations associated with this system can be found in the Hussain paper and are not applicable to the system designed and fabricated in this report. For the system design presented in this report, the stiffness of the joints used sat somewhere in between extremely flexible and very stiff based upon testing results. The gripping system developed in this report was very compliant in that it could wrap around large objects, but also stiff enough that it could hold objects at its tips in a pinching configuration.

The initial prototype developed consisted of a pancake configuration NEMA 17 with a custom 3D printed single start worm. The worm was then mated with a custom worm gear that also contained pulley extrusions. A worm drive was chosen for this system for its innate ability to not be back drivable. This allows for the system to maintain a grasp on a target object without consuming power, which would prove to be a beneficial property for integration into a powered prosthetic system where battery consumption rates are of large importance. The gripping surfaces of the system were two finger-like apparatuses that were 18 mm apart at the base and 40 degrees apart from each other in the resting position.

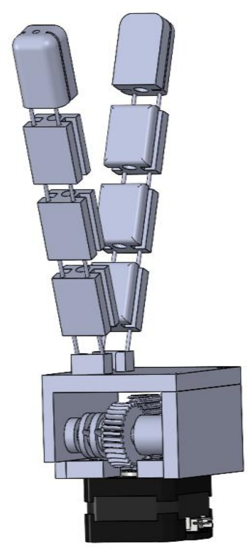

Figure 2. Actuating Grasping System Revision #1

The spool gear and the worm gear had a gear reduction of 30:1. The motor was able to actuate the dual-spool gear, but it was incapable of holding objects exceeding 0.25 lbs. One of the grippers did not function since it did not spool properly during testing. Furthermore, the gear would skip steps from time to time due to misalignment and wearing of the 3D printed gears. Adjustments were made to organize the spooling and improve the gear ratio.

In the second iteration, the system shifted from a dual-spool worm gear with 30 teeth to two single-spool worm gears with 60 teeth each. The distance and angle between the grippers remained the same as the first iteration. Due to the change in teeth for the two gears surrounding the worm gear, the system attained a 60:1 gear reduction. Although both fingers functioned properly, the gripper was once again only capable of holding up to 0.25 lbs. indicating the NEMA 17 did not provide enough torque to meet the project's requirements. To accommodate for the lack of torque, the NEMA 17 motor was switched out with a full size NEMA 17, which allowed the system's carrying capacity to increase up to 0.75 lbs.

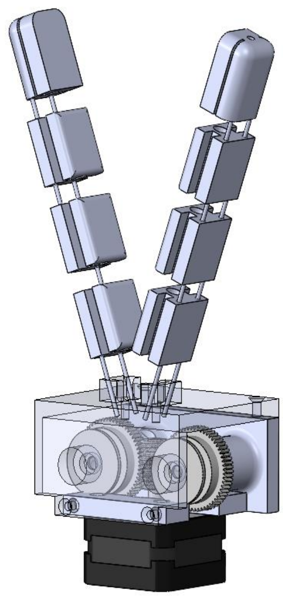

Figure 3. Actuating Grasping System Revision #2

The motor, however, began to skip steps again. The NEMA 17 motor, for both the lower pancake and normal configurations just did not have enough torque to drive the system. As a result, adjustments were made to eliminate the need for spool gears and a switch from a stepper motor to a high torque DC motor with a built-in gearbox was made.

For the final iteration of the soft-rigid gripper, a Greartisan's high torque, dual-shaft DC motor was used to eliminate the need for a 3D printed worm drive. The gap between the two fingers increased to 45 mm and the angle remained the same. It should also be noted that the frictional losses related to the path of the cable was reduced to a minimum as each individual cable was aligned along the tangent of the spool. In the two previous iterations, the cable would exit from the spool and out towards the tips of the fingers at non-optimal angles and with a lot of interference with the surfaces around it.

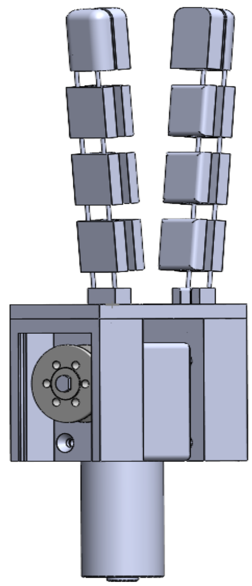

Figure 4. Actuating Grasping System Revision #3

This system ultimately was capable of grasping objects up to 3 lbs. which greatly exceeded the required amount of 1 lb. for this project. The motor exerted a torque of 3.92 N-m, which was enough to have both fingers bend over and on top of each other by crossing the midplane between the two.

TACTILE SENSING

The requirements of this project state that gripper shall be able to automatically close around objects of varying size, weight, and material without damaging or dropping the object. To accomplish this, a force sensing system was implemented to provide a means for output force regulation. This was done via magnetometer and permanent magnet displacement. Two silicone fingertips embedded with permanent magnets were designed, molded, and placed over the 3D printed fingertip joint segments. Each fingertip joint was paired with a 3-axis magnetometer to allow force sensing via displacement of the permanent magnets within the silicone tips.

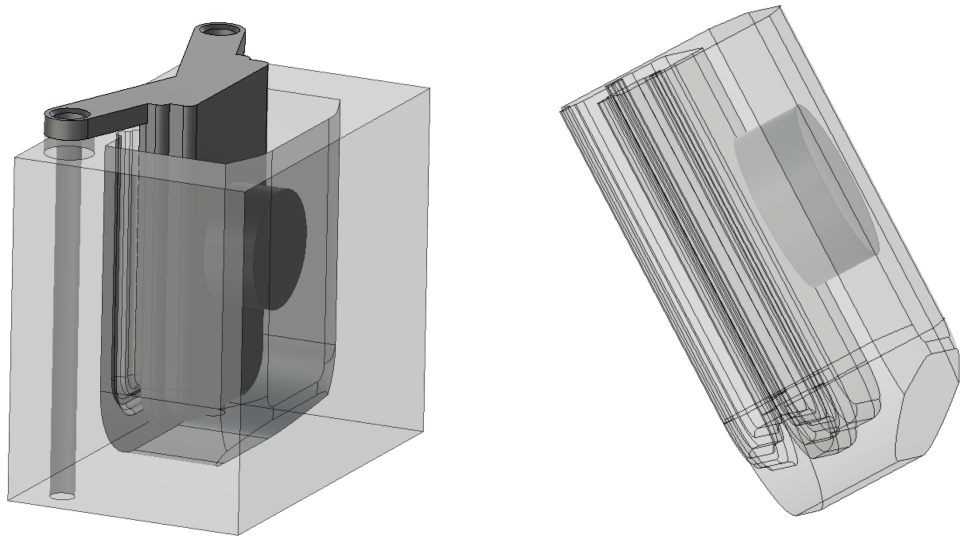

Figure 5. Two-piece fingertip mold (Left) , Silicone fingertip embedded with magnet (Right)

Testing was performed on magnets of various sizes, and it was determined that a 12mm diameter x 5mm thick permanent magnet provided a magnetic field that optimally affected the sensor while still fitting within the profile of the fingertip design. Oomoo's Shore 37A Sorta Clear silicone was chosen for this project due to availability. The relatively high hardness of this silicone worked well for this project. It allowed for some compliance, provided a high amount of friction between itself and the surface it was interacting with, and it did not sag under its own weight. The silicone mold for the fingertip sensor was 3D printed and composed of an outer and inner piece that fastened together via two M3x0.5 mm screws.

The silicone fingertip was designed with a compressible gap between it and the 3D printed joint segment. This allowed for two separate sensitivities of force, a force sensitivity for "soft" objects, when compressing the open gap, and a force sensitivity for harder objects when compressing the silicone against the joint segment. In practice, sensing soft objects was very effective, but with heavier objects, the magnet could not be displaced past a certain amount. This led to "heavy" objects usually being grasped by reaching a preconfigured motor time-out period.

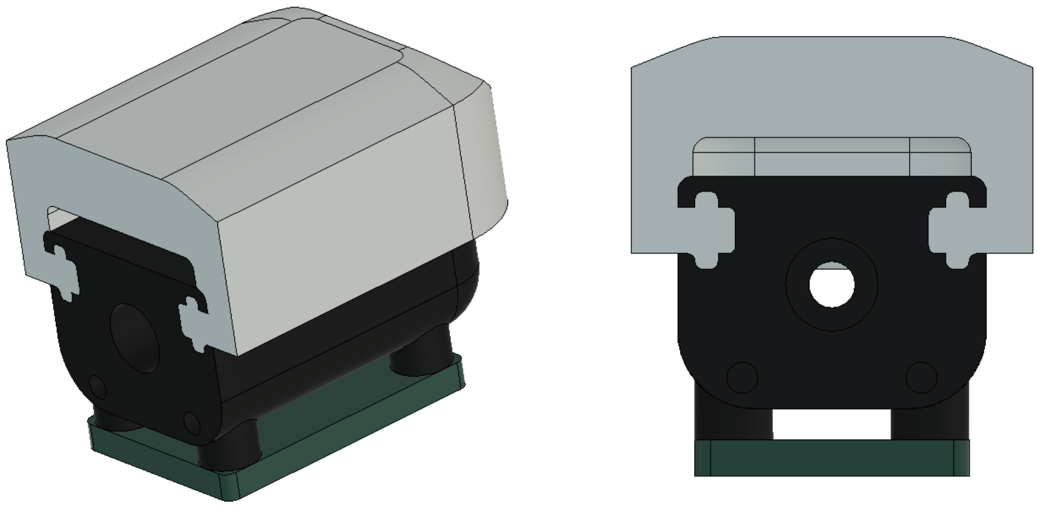

Figure 6. Fingertip Assembly CAD Model

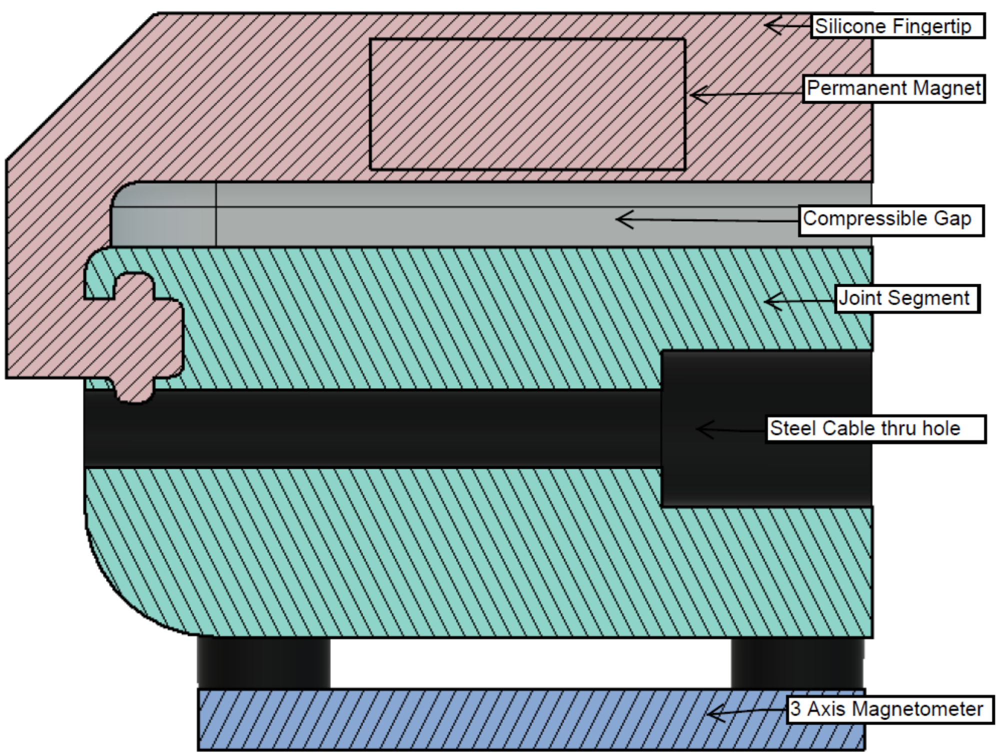

Figure 7. Fingertip assembly cross section view

Additionally, silicon pads were designed and molded for the other segments of the fingers. These pads increased the gripping power of the fingers by increasing the surface area in contact with the target object. To add to this, since the pads were made of a silicone elastomer, they possessed a high coefficient of friction to prevent slipping between the target object and the gripping surface.

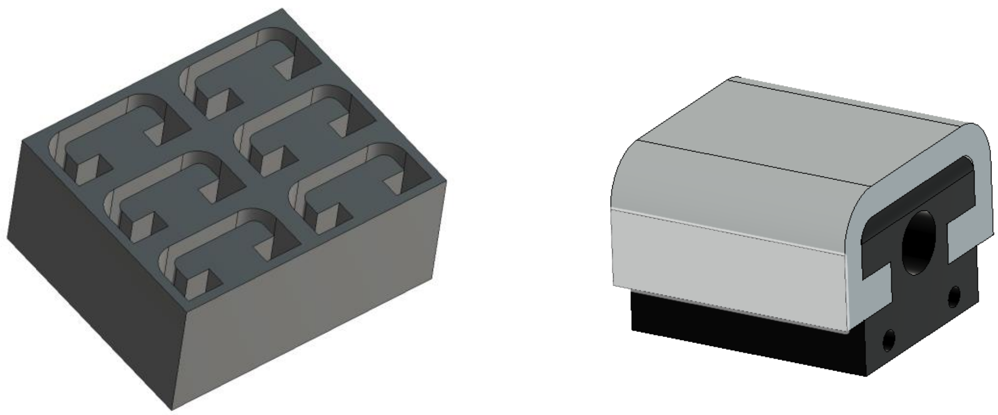

Figure 8. 3D printed mold for silicone joint covers (Left) and joint and silicone assembly (right)

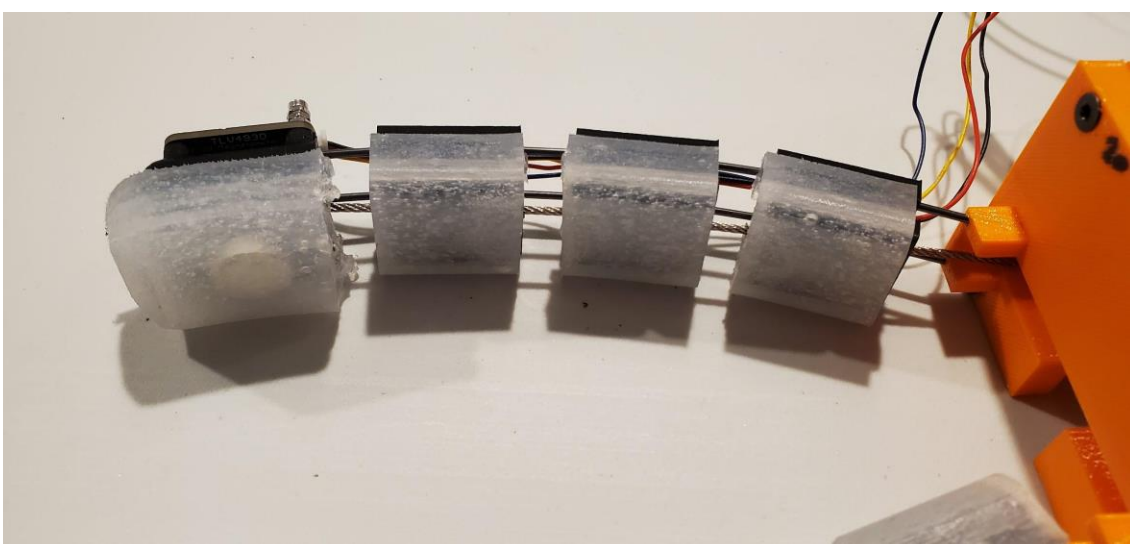

Figure 9. A Finger assembled with silicone pads, magnet, and sensor

SOFTWARE

The gripper requirements state that the system shall be able to grip objects of differing sizes, weights, and materials and to do so autonomously. This means a method of force output feedback is necessary to communicate to the gripper when it has gripped an object with sufficient clamping force to not drop the object, but not so hard that it damages the object. The chosen method of feedback was a custom-built tactile sensor system embedded in both fingertips. A 3-axis magnetometer was paired with a permanent magnet and used to create the tactile sensor. The magnetometer communicates over I2C, which is convenient for connecting more than one sensor. All that is needed to add more sensors is to change the address for each sensor. Utilizing timer interrupts attached with callback functions ensured that the accelerometer data and any actions taken were executed in real-time. This led to a very responsive grip and measuring system where the gripper can quickly respond to changes in applied force and avoid damaging the object. Furthermore, several helper functions and a basic Serial UI were implemented to speed up testing.

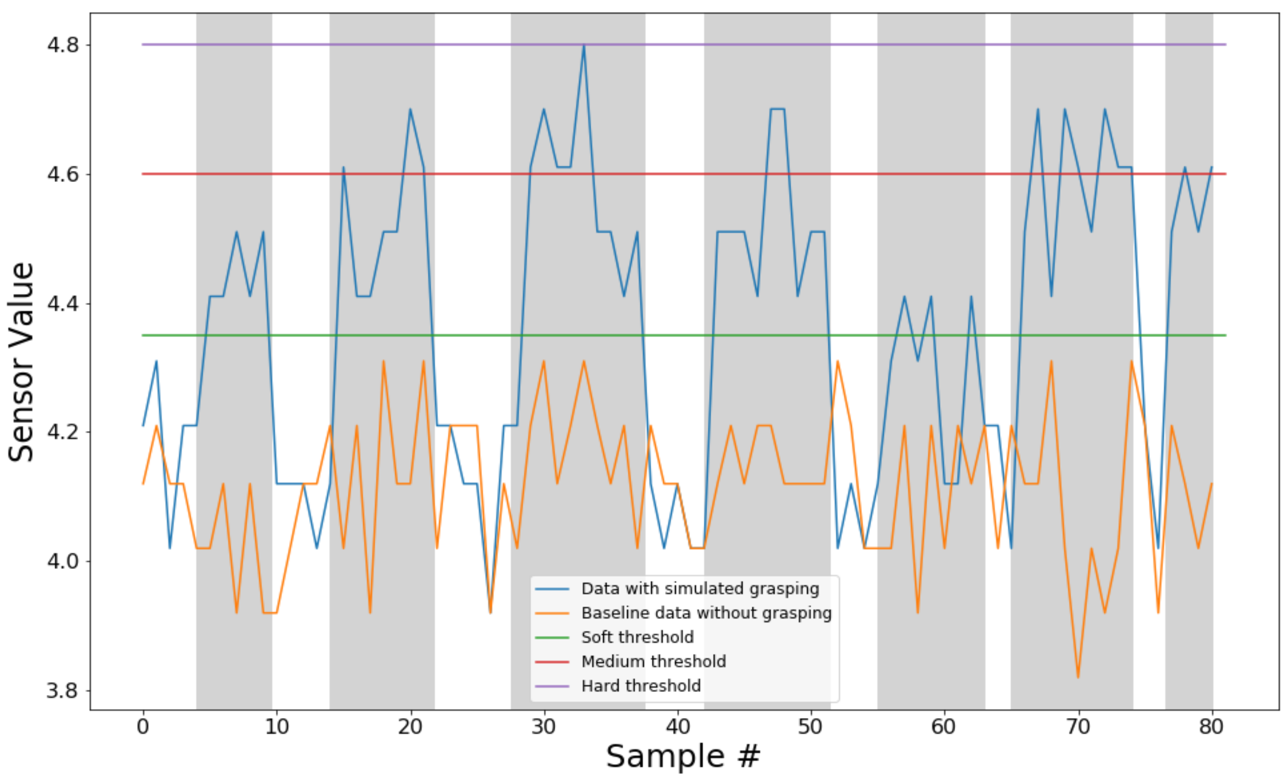

Figure 10. Baseline and test grasping data with thresholding, gray bars indicate grasping events

Data was logged via a python script that would read in serial data and write it out to a text file. The data in Figure 10 is a summary of the sensor's readings both with and without applied forces. Two data files were written, one recorded baseline data, meaning no forces were imparted on the sensor and the other file included data with force sensing. Those two data files were used to profile a no-load condition and profile noise to determine the minimum displacement required to produce a true force reading. From that data, three thresholding profiles were created, one for soft objects, medium objects, and hard objects. Additionally, a failsafe timer was implemented so that if the sensor did not get a good reading, the motor would stop after a certain period of run-time.

Materials and Manufacturing

The bulk of the gripper was made of 3D printed parts. The fingers, baseplate, spools and motor mount were all made of standard PLA. A Prusa Mini and Prusa I3 MK3S 3D printer were used to print the custom parts. A 0.4mm nozzle was used with 0.2mm layer height and 15% rectilinear infill for all parts. There was one issue where the baseplate cracked under high load, so the baseplate was reprinted with six perimeters instead of two and a 30% rectilinear infill to strengthen the part.

Shore 37A silicone was used for creating the fingertip and joint pads. SORTA-Clear™ 37 Clear Silicone Rubber by Smooth-On was used as due to its availability and the material stiffness was ideal for this application. The silicone came as a two-part mixture. To create the mold, 1A:1B by volume, or mass, had to be mixed. The proper mixing amounts were determined using a digital scale. It was critical to evenly combine and stir the two parts together to avoid uncured silicone residue left on the mold. During the molding process, one of the silicone fingertip did not cure long enough and was stickier than expected. All molds were designed, and 3D printed for this project.

The compliant material used was 0.06” diameter Super-Elastic Nitinol Wire ASTM F2063, purchasable from McMaster-Carr. This material was then cut into equal lengths of approximately 25mm using cable shears. A key factor in preventing the fingers from closing unevenly and without twisting was cutting these rods to very exact lengths, or as close to this as possible with the methods available In a future revision, a new method for cutting the rods would be advised, possibly with a saw or another more precise method. Furthermore, Nitinol proved to be tougher than anticipated as it would dig into the 3D printed joints causing an imbalance.

Being a tendon-driven gripper, one of the key materials used in the system manufacturing was steel cable for the tendons. A 0.0625” (1.58 mm) diameter steel cable was settled on as it was the right combination of strength and flexibility. Initially a 0.039” (1 mm) diameter cable was used, but in manufacturing the first two revisions, no more was available at local hardware shops. A 0.09375” (2.38 mm) diameter cable was also used, but it was too stiff for the system. Cutting the cable to the same length and spooling it the same amount on both sides proved to be challenging. It involved starting with excess cable and tightening the fingers to make appropriate cuts. This process would be repeated until it was determined the tension on both fingers were relatively the same. One way of fixing this issue would be to use one motor for each finger, but that would add weight and bulk to the gripper. Alternatively, cutting the cable wires the same length before putting everything together would have contributed to a more balanced gripper. Regardless, there may be another method of manufacturing the fingers to have precise amounts of equal tension on each finger.

Other components included the motor, motor driver, magnetometer, and microcontroller. The final motor used was a 12V DC motor with a 1:634 worm gearbox with dual shaft output purchased through Amazon (See Appendix B). An Adafruit DRV8871 DC Motor Driver Breakout Board was used to drive the motor. This motor driver could provide 6.5V-43V with 3.6A peak current and was more than sufficient to power our motor. The magnetometer used was the Adafruit TLV493D Triple-Axis Magnetometer. This magnetometer was reused from a previous project and provided 3-axis data, meaning one could read perpendicular forces as well as shear. The microcontroller used was the Arduino Nano 33 IOT. This board was small and contained all the necessary communication buses, PWM, digital and analog pins needed to control the gripper. Additionally, it was both WiFi and Bluetooth enabled if a future revision of the gripper was to be controlled wirelessly.

Analysis and Results

Testing was performed on several objects of differing weight, size, and material. A 32 oz water bottle was used to test both the gripping strength and the gripper's compliance.

Figure 11. Images of the gripper holding a water bottle weighing 1.87 pounds

In Figure 11, the steps taken for the water bottle test along with the results are shown. The water bottle was filled to approximately 24 oz, leading to a total weight of 1.87 pounds, well above the 1 lbs. requirement for the system. It can be seen that the system successfully grasps the object. In another test that is not documented photographically in this report, the gripper managed to hold a 3.5 lbs. drill. These photos were omitted as holding the drill caused extreme bending in the finger joints. Although the gripping system held the drill, the extreme bending presents a great area of concern for deeming it a successful test. Testing was also performed on lighter and more sensitive objects to verify that the system could satisfy the other design requirements. A potato chip bag and an empty plastic cup were used to test the lower bounds of the tactile sensor. A successful test would mean that the system was able to autonomously grasp and sense the force it is outputting onto an object of low stiffness to avoid causing damage. In Figure 12 and Figure 13, this test is put on display.

Figure 12. Unopened potato chip bag weighing 0.16 pounds (approx. 72.5 grams)

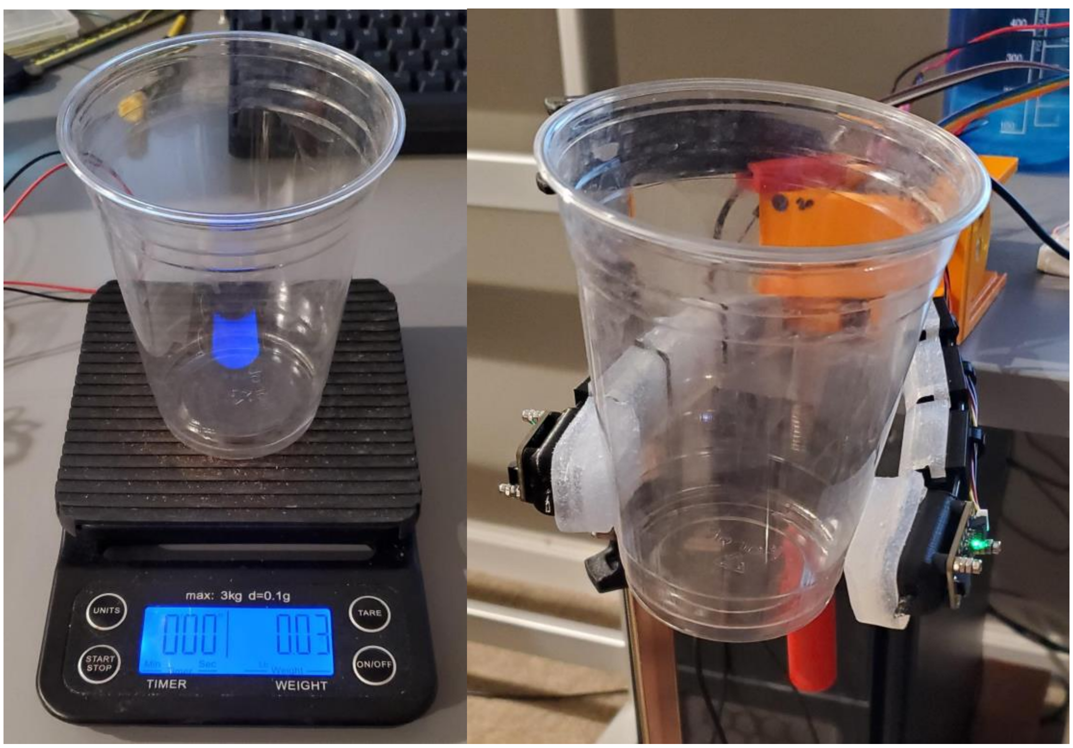

Figure 13. Plastic cup weighing 0.03 pounds (approx. 13.6 grams)

The chip bag amongst the lighter objects used was the one with the least amount of stiffness as its volume is mostly taken up by air. The gripper, however, managed to grasp the chip bag autonomously and securely without losing hold or damaging the bag by popping. For the plastic cup, the area of concern was that the cup was both stiff and easily deformable due to its thin walls. Too large of a force would deform the cup, but for the system to stop actuating, it needed to detect a normal force to trigger the system. The tactile sensor system, however, performed as designed and registered the slightest bit of displacement in the magnet at the sensing surface. It should be noted that in three grasping tests done, all done autonomously and executed in under the 8 second limit. The results of these tests were successful and proved that our system met all the design requirements.

Conclusion

The design was successful in achieving the goals stated in the project prompt. The gripper had minimal issues gripping the three objects used for the in-class test. There were no issues gripping the one-pound water bottle, while the balloon and aluminum-leaf attached to an empty water bottle only had minor issues with sensing. Careful placement of the test object was necessary to ensure the sensors would come into contact and sense the appropriate amount of force, but when the object was placed correctly, the gripper was successful. There is room for improvement, particularly in the placement and quantity of sensors. If sensors were placed evenly throughout the fingers, the gripper would have a wider range of gripping orientations that could sense force. Currently, the gripper can only sense forces if the object is pinched between the fingertips or if the fingers wrap around the object. By adding sensors to the joints, the gripper could use only the joint members to grip and sense, opening the range of object shapes and sizes that could be sensed. Additionally, the manufacturing process has significant room for improvement. Cutting precise lengths for both the nitinol rods and the steel cable would have streamlined assembly and testing. Certain elements, particularly the gripper's base plate and spool, would see the greatest benefit if they were machined from aluminum as they showed wear at higher loads. Lastly, a deeper look into algorithms that could be used to improve the gripper's force sensing profile. By encoding a wider variety of objects and materials in addition to adding more sensors, the gripper would be more universal. In conclusion, while there is room for improvement, for the time given to complete this project, the gripper developed exhibited incredible success.

References

[1] L. Yuan, J. Li, L. Hong, M. Dong. "A Systematic Analysis of Hand Movement Functionality: Qualitative Classification and Quantitative Investigation of Hand Grasp Behavior". *Frontiers in Neurorobotics*. https://www.frontiersin.org/article/10.3389/fnbot.2021.658075. Accessed 9 Apr. 2022.

[2] R. P. Rocha, P. A. Lopes, A. T. de Almeida, M. Tavakoli, C. Majidi. Fabrication and characterization of bending and pressure sensors for a soft prosthetic hand. Journal of Micromechanics and Microengineering. 28(2018).

[3] P. Saccomandi, E. Schena, C. M. Oddo, L. Zollo, S. Silverstri, E. Guglielmelli, Microfabricated Tactile Sensors for Biomedical Applications: A Review, Biosensors (Basel). 2014 Nov 3.

[4] Hussain, Irfan & Malvezzi, Monica & Gan, Dongming & Iqbal, Muhammad Zubair & Seneviratne, Lakmal & Prattichizzo, Domenico & Renda, Federico. (2020). Compliant gripper design, prototyping, and modeling using screw theory formulation. The International Journal of Robotics Research. 40. 027836492094781. 10.1177/0278364920947818.

Appendix A

CAD Models and Photos

A.1 – Hussain/Khalifa University Gripping System

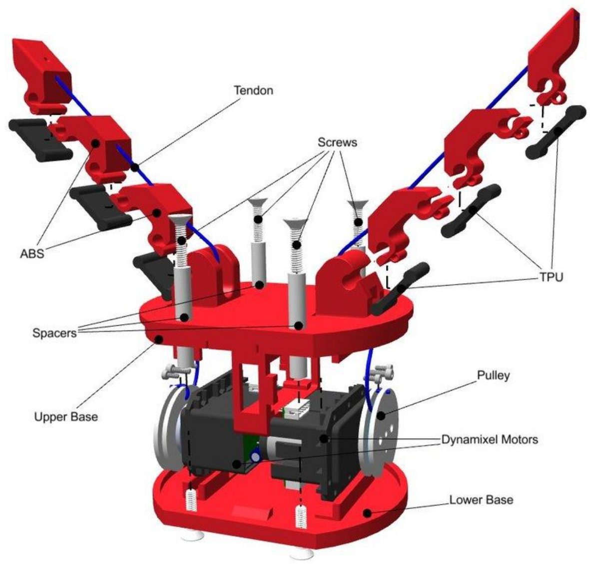

Figure 1. Compliant tendon driven system from Khalifa University

A.2 – Chao, Rodriguez, Weigel Gripping System Revision #1

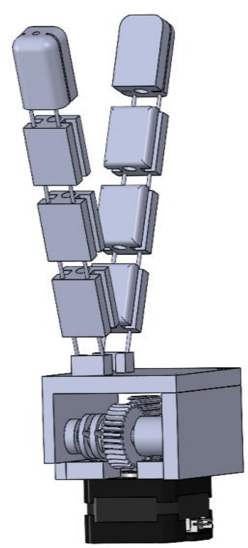

Figure 2. Actuating Grasping System Revision #1

Appendix A

CAD Drawings and Photos

A.3 – Chao, Rodriguez, Weigel Gripping System Revision #2

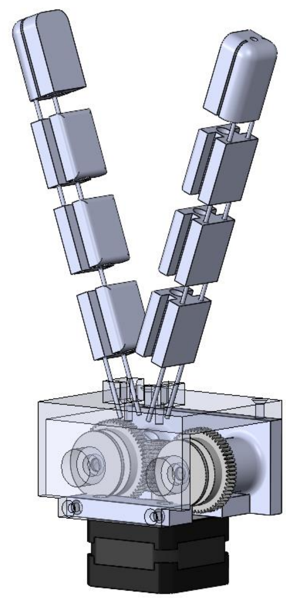

Figure 3. Actuating Grasping System Revision #2

A.4 – Chao, Rodriguez, Weigel Gripping System Revision #3

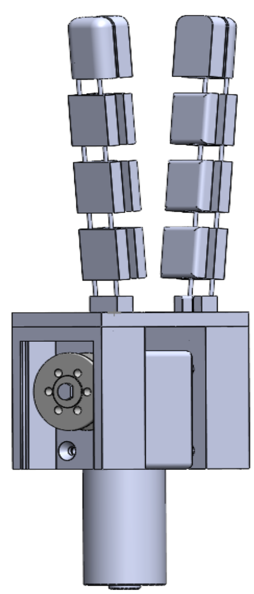

Figure 4. Actuating Grasping System Revision #3

Appendix B

Materials

B.1 Material Table

Part	Category	Purchase Price	Quantity or amount able to be made from Purchase	Price/Single Unit	Distributor
Greartisan DC 12V 10RPM 40Kg.cm 8mm Double Shafts Self-Locking Reversible Worm Gear Motor with Cable, High Torque Speed Reduction Motor, Turbine Electric Gearbox Motor	Motor	$28.99	1	$28.99	Amazon
Adafruit DRV8871 DC Motor Driver Breakout Board - 3.6A Max	Motor Driver	$10.94	1	$10.94	Adafruit
1/16 " Steel Cable	Tendon	$3.70	5	$0.37	Home Depot
SORTA-Clear™ 37 Clear Silicone Rubber - Trial Unit	Silicone	$44.22	1	$44.22	Smooth-0n
Arduino Nano 33 IoT	Microcontroller	$18.40	1	$18.40	Arduino
12V 2A Power Supply Adapter, SANSUN 120VAC to 12VDC Transformer with 5.5x2.5mm DC Output Jack, 12 Volt 2 Amp	Power Supply	$7.99	1	$7.99	Amazon
Adafruit TLV493D Triple-Axis Magnetometer	Magnetometer	$11.9	2	$5.95	Adafruit
High-strength 'rare earth' magnet 12.7mm x 5mm	Magnet	$5.00	2	$2.50	Adafruit
.06" Nitinol Rod, 2ft sections qty: 5 - 8320K42	Structural	$136.86	5	$27.37	McMaster-Carr
			Total Cost =	$147.10	

Table 1. Bill of Materials

Appendix C

Electronics

C.1 Wiring and Components

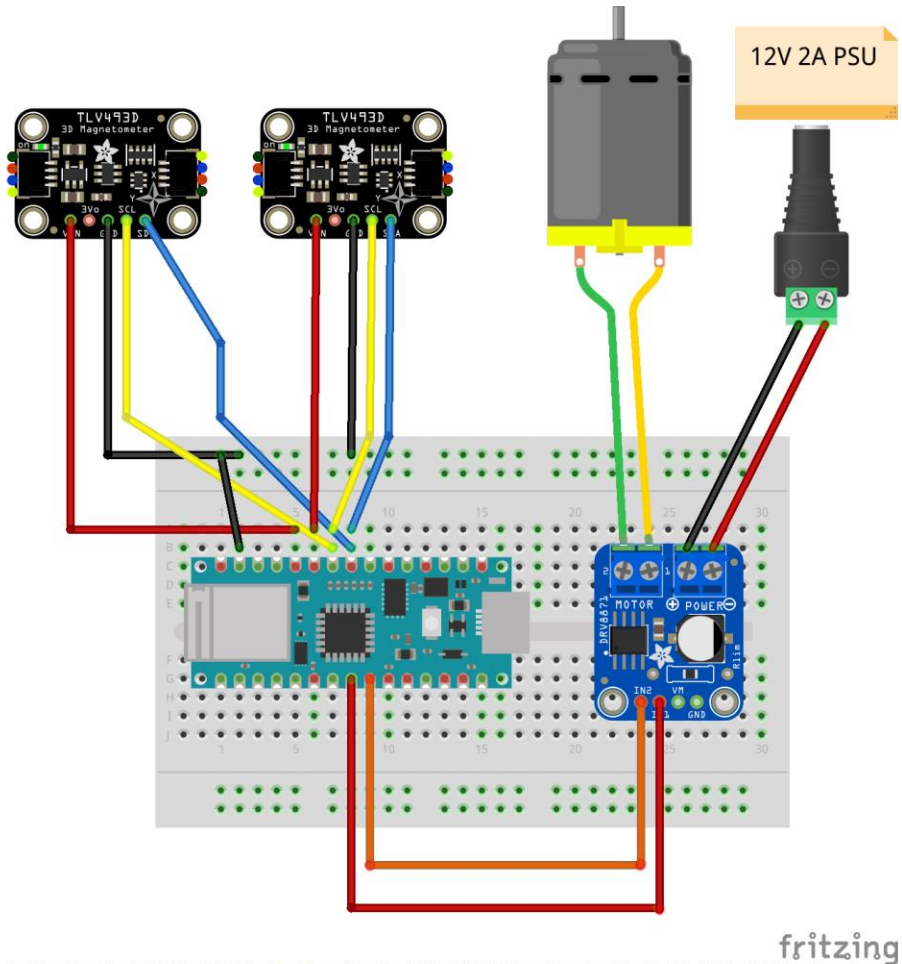

Figure 14. Component layout and wiring diagram

Appendix D

Software

D.1 Arduino Code

```
// Written By Cameron Weigel for ME 5741 Compliant Gripper project #2
// Updated 4/6/2022

#include <ArduinoBLE.h>
#include <Tlv493d.h>
#include <Wire.h>
#include "Ticker.h"

#define POWER_PIN1 20
#define POWER_PIN2 21

#define motorOpen 3 // IN2
#define motorClose 2 // IN1

Tlv493d Tlv493d_Sensor1 = Tlv493d();  //Sensor 1 shall have I²C address 1
Tlv493d Tlv493d_Sensor2 = Tlv493d();  //Sensor 2 shall have I²C address 2

char incomingByte = '\n';
int motorTime = 500; // miliseconds
int motorTimeSmall = 250;

// Combined implementation of motor state

float SOFT1= 4.35;
float MEDIUM1 = 4.6;
float HARD1 = 5.0;

float SOFT2= 4.35;
float MEDIUM2 = 4.6;
float HARD2 = 4.9;


float hardness1 = SOFT1;
float hardness2 = SOFT2;

bool motorState = true;
bool setupState = true;

void checkSensors();
void timeOver();
void printSensors();
```

```
Ticker sensorTimer(checkSensors, 50,0,MILLIS);
Ticker maxTimer(timeOver, 1750,0,MILLIS);
Ticker sensorPrint(printSensors, 1000,0,MILLIS);

void setup() {
  Serial.begin(9600);

  pinMode(motorOpen, OUTPUT);
  pinMode(motorClose, OUTPUT);

  pinMode(POWER_PIN1, OUTPUT);
  pinMode(POWER_PIN2, OUTPUT);
  digitalWrite(POWER_PIN1, LOW);
  digitalWrite(POWER_PIN2, LOW);


  delay(100);

  digitalWrite(POWER_PIN2, HIGH);
  delay(150);
  //Configure sensor 2 to address 2 at first, so it will not respond when
later address 1 is configured.
  Tlv493d_Sensor2.begin(Wire, TLV493D_ADDRESS2, true);

  //power up and configure sensor 1
  digitalWrite(POWER_PIN1, HIGH);
  delay(150);
  //Configure sensor 1 to address 1.
  //IMPORTANT: Don't perform an I²C sensor reset (last bool argument) as this
would also reset sensor 2
  //           -> Then sensor 2 would again respond to address 1, what we need
to avoid.
  Tlv493d_Sensor1.begin(Wire, TLV493D_ADDRESS1, false);

  sensorTimer.start();
  sensorPrint.start();


}


void loop() {
    sensorTimer.update();
    sensorPrint.update();
    //maxTimer.update();

  if (Serial.available() > 0) {
```

```
    if (setupState)
    {
       Serial.println("Setup - Enter S for soft, M for medium, H for hard");
       while (setupState)
      {
        incomingByte = Serial.read();

        if (incomingByte == 'S')
        {
          hardness1 = SOFT1;
          hardness2 = SOFT2;
          setupState = false;
          Serial.println("SOFT hardness chosen");
        }
        if (incomingByte == 'M')
        {
          hardness1 = MEDIUM1;
          hardness2 = MEDIUM2;
          setupState = false;
          Serial.println("MEDIUM hardness chosen");
        }
        if (incomingByte == 'H')
        {
          hardness1 = HARD1;
          hardness2 = HARD2;
          setupState = false;
          Serial.println("HARD hardness chosen");
        }
      }
    }

    incomingByte = Serial.read();
    if (incomingByte != '\n')
    {
      Serial.print("I received: ");
      Serial.println(incomingByte);
      if (incomingByte == 'C') {
        // Close
        Serial.println("Closing");
        close(motorClose, motorTime);
      }

      if (incomingByte == 'O') {
        // Open
        Serial.println("Opening");
        open(motorOpen, motorTime);
      }

            Serial.print("I received: ");
```

```
      Serial.println(incomingByte);
      if (incomingByte == 'c') {
        // Close
        Serial.println("Closing - Small");
        close(motorClose, motorTimeSmall);
      }

      if (incomingByte == 'o') {
        // Open
        Serial.println("Opening - Small");
        open(motorOpen, motorTimeSmall);
      }

      if (incomingByte == 'R') {
        Serial.println("Running motor");
        while (motorState) {
          if (maxTimer.state() == STOPPED || maxTimer.state() == PAUSED)
          {
            maxTimer.resume();
          }
          sensorTimer.update();
          maxTimer.update();
          run(motorClose);
        }
        Serial.println("Stopping motor");
        stop(motorClose);
        motorState = true;
      }
      if (incomingByte == 'Z')
      {
        setupState = true;
      }
      if (incomingByte == 'H')
      {
        Serial.println(" Help Section: ");
        Serial.println("C : close motor for 500ms");
        Serial.println("O : open motor for 500ms");
        Serial.println("R : run motor until sensors read hardness value");
        Serial.println("Z : reset setup to choose different hardness value");
      }
    }
  }

}


void close(int closePin , int milisec)
{
  analogWrite(closePin, 255);
  delay(milisec);
```

```
  analogWrite(closePin, 0);
}

void open(int openPin, int milisec)
{
  analogWrite(openPin, 255);
  delay(milisec);
  analogWrite(openPin, 0);
}

void stop(int motorPin)
{
  analogWrite(motorPin, 0);
}

void run(int motorPin)
{
  analogWrite(motorPin, 255);
}

// Combined Implementation of motorState

void checkSensors() {
  Tlv493d_Sensor1.updateData();
  Tlv493d_Sensor2.updateData();
  if ((abs(Tlv493d_Sensor1.getZ()) >= hardness1) ||
(abs(Tlv493d_Sensor2.getZ()) >= hardness2))
  {
    motorState = false;
    maxTimer.stop();
  }
  else
  {
    motorState = true;
  }
}

void printSensors()
{
  Serial.print("Sensor 1: ");
  Serial.println(Tlv493d_Sensor1.getZ());
  Serial.println(Tlv493d_Sensor1.getX());
  Serial.println(Tlv493d_Sensor1.getY());
  Serial.print("Sensor 2: ");
  Serial.println(Tlv493d_Sensor2.getZ());
  Serial.println(Tlv493d_Sensor2.getX());
  Serial.println(Tlv493d_Sensor2.getY());
}
```

```
void timeOver() {
  maxTimer.stop();
  motorState = false;
  Serial.println("Max timeout reached, stopping motor");
}
```

D.2 Python Code

```
#!/usr/local/bin/python3

# Python script used for writing data files over serial
import serial, io
device = "/dev/ttyACM0"
baud = 9600
filename = 'gripper-log.txt'

with serial.Serial(device,baud) as serialPort, open(filename,'wb') as outFile:
    while(1):
        line = serialPort.readline()
        print(line)
        outFile.write(line)
        outFile.flush()
```

```
#!/usr/bin/env python
# coding: utf-8

# In[55]:


## Written by Cameron Weigel 4/8/2022 for ME 5741 Project #2 -
Biomechanically Inspired Gripper -
#
# Imports two log files, one for data including grasping of objects and the
other of baseline data to establish
# thresholding to determine if the object is grasped without damage or
slippage
#
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle

data = pd.read_csv('gripper-log.txt',sep='\n', header=None)
dataBaseline = pd.read_csv('gripper-log2.txt',sep='\n', header = None)
data = pd.DataFrame(data)
dataBaseline = pd.DataFrame(dataBaseline)
dataBaseline = dataBaseline.truncate(after=data.size-1)
```

```
# In[155]:


# Plotting Routines

fig, ax = plt.subplots()
fig.set_size_inches(16,10)
x1 = np.linspace(0,data.size,data.size)
y1 = np.empty(data.size)
y1.fill(4.35)

x2 = np.linspace(0,data.size,data.size)
y2 = np.empty(data.size)
y2.fill(4.6)

x3 = np.linspace(0,data.size,data.size)
y3 = np.empty(data.size)
y3.fill(4.8)


ax.plot(-data[0])
ax.plot(-dataBaseline[0])

ax.plot(x1,y1)
ax.plot(x2,y2)
ax.plot(x3,y3)

LARGE_TEXT = 28
MEDIUM_TEXT = 18
SMALL_TEXT = 12

plt.rc('axes', labelsize=24)    # fontsize of the x and y labels
plt.rc('xtick', labelsize=MEDIUM_TEXT)    # fontsize of the tick labels
plt.rc('ytick', labelsize=MEDIUM_TEXT)    # fontsize of the tick labels
plt.rc('legend', fontsize=SMALL_TEXT)    # legend fontsize


plt.xlabel('Sample #')
plt.ylabel('Sensor Value')


legend_desc = ['Data with simulated grasping','Baseline data without
grasping','Soft threshold', 'Medium threshold', 'Hard threshold']
plt.legend(legend_desc,loc='best')

ax.add_patch(Rectangle((4, 2), 5.6, 3,color="lightgray"))
```

```
ax.add_patch(Rectangle((14, 2), 7.7, 3,color="lightgray"))

ax.add_patch(Rectangle((27.5, 2), 10, 3,color="lightgray"))

ax.add_patch(Rectangle((42, 2), 9.4, 3,color="lightgray"))

ax.add_patch(Rectangle((55, 2), 4.7, 3,color="lightgray"))

ax.add_patch(Rectangle((55, 2), 8, 3,color="lightgray"))

ax.add_patch(Rectangle((65, 2), 9, 3,color="lightgray"))

ax.add_patch(Rectangle((76.5, 2), 3.5, 3,color="lightgray"))

plt.show()
```